A New Lower Bound on the Maximum Number of Satisfied Clauses in Max-SAT and its Algorithmic Applications *

Robert Crowston, Gregory Gutin, Mark Jones, Anders Yeo
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
robert,gutin,markj,anders@cs.rhul.ac.uk

Abstract

A pair of unit clauses is called conflicting if it is of the form (x), $(\bar{x})$. A CNF formula is unit-conflict free (UCF) if it contains no pair of conflicting unit clauses. Lieberherr and Specker (J. ACM 28, 1981) showed that for each UCF CNF formula with m clauses we can simultaneously satisfy at least $\hat{\varphi}m$ clauses, where $\hat{\varphi} = (\sqrt{5}-1)/2$. We improve the Lieberherr-Specker bound by showing that for each UCF CNF formula F with m clauses we can find, in polynomial time, a subformula F' with m' clauses such that we can simultaneously satisfy at least $\hat{\varphi}m + (1-\hat{\varphi})m' + (2-3\hat{\varphi})n''/2$ clauses (in F), where n'' is the number of variables in F which are not in F'.

We consider two parameterized versions of MAX-SAT, where the parameter is the number of satisfied clauses above the bounds $m/2$ and $m(\sqrt{5}-1)/2$. The former bound is tight for general formulas, and the later is tight for UCF formulas. Mahajan and Raman (J. Algorithms 31, 1999) showed that every instance of the first parameterized problem can be transformed, in polynomial time, into an equivalent one with at most $6k+3$ variables and $10k$ clauses. We improve this to $4k$ variables and $(2\sqrt{5}+4)k$ clauses. Mahajan and Raman conjectured that the second parameterized problem is fixed-parameter tractable (FPT). We show that the problem is indeed FPT by describing a polynomial-time algorithm that transforms any problem instance into an equivalent one with at most $(7+3\sqrt{5})k$ variables. Our results are obtained using our improvement of the Lieberherr-Specker bound above.

*A preliminary version of this paper appeared in the Proceedings of IPEC 2010. Research was supported in part by an International Joint grant of Royal Society. Research of Gutin was also supported in part by the IST Programme of the European Community, under the PASCAL 2 Network of Excellence.

1 Introduction

Let $F = (V, C)$ be a CNF formula, with a set V of variables and a multiset C of non-empty clauses, $m = |C|$ (i.e., m is the number of clauses in C; each clause is counted as many times as it appears in C), and $\mathrm{sat}(F)$ is the maximum number of clauses that can be satisfied by a truth assignment. With a random assignment of truth values to the variables, the probability of a clause being satisfied is at least 1/2. Thus, $\mathrm{sat}(F) \geq m/2$ for any F. This bound is tight when F consists of pairs of *conflicting unit clauses* (x) and $(\bar{x})$. Since each truth assignment satisfies exactly one clause in each pair of conflicting unit clauses, it is natural to reduce F to the *unit-conflict free (UCF)* form by deleting all pairs of conflicting clauses. If F is UCF, then Lieberherr and Specker [11] proved that $\mathrm{sat}(F) \geq \hat{\varphi}m$, where $\hat{\varphi} = (\sqrt{5} - 1)/2$ (golden ratio inverse), and that for any $\epsilon > 0$ there are UCF CNF formulae F for which $\mathrm{sat}(F) < m(\hat{\varphi} + \epsilon)$. Yannakakis [19] gave a short probabilistic proof that $\mathrm{sat}(F) \geq \hat{\varphi}m$ by showing that if the probability of every variable appearing in a unit clause being assigned TRUE is $\hat{\varphi}$ (here we assume that for all such variables x the unit clauses are of the form (x)) and the probability of every other variable being assigned TRUE is 1/2, then the expected number of satisfied clauses is $\hat{\varphi}m$.

A formula $F' = (V', C')$ is called a *subformula* of a CNF formula $F = (V, C)$ if $C' \subseteq C$ and V' is the set of variables in C'. If F' is a subformula of F then $F \setminus F'$ denotes the subformula obtained from F by deleting all clauses of F'. A formula $F = (V, C)$ is called *expanding* if for each $X \subseteq V$, the number of clauses containing at least one variable from X is at least $|X|$ [17]. It is known (this involves so-called matching autarkies, see Section 2 for details) that for each CNF formula $F = (V, C)$ a subformula $F' = (V', C')$ can be found in polynomial time such that $\mathrm{sat}(F) = \mathrm{sat}(F \setminus F') + |C'|$ and the subformula $F \setminus F'$ is expanding. In this paper, the main technical result is that $\mathrm{sat}(F) \geq \hat{\varphi}|C| + (2 - 3\hat{\varphi})|V|/2$ for every expanding UCF CNF formula $F = (V, C)$. Combining this inequality with the previous equality for $\mathrm{sat}(F)$, we conclude that for each UCF CNF formula $F = (V, C)$ a subformula $F' = (V', C')$ can be found in polynomial time such that

$$\mathrm{sat}(F) \geq \hat{\varphi}|C| + (1 - \hat{\varphi})|C'| + (2 - 3\hat{\varphi})|V \setminus V'|/2.$$

The last inequality improves the Lieberherr-Specker lower bound on $\mathrm{sat}(F)$.

Mahajan and Raman [14] were the first to recognize both practical and theoretical importance of parameterizing maximization problems above tight lower bounds. (We give some basic terminology on parameterized algorithms and complexity in the next section.) They considered MAX-SAT parameterized above the tight lower bound $m/2$:

SAT-A$(m/2)$
Instance: A CNF formula F with m clauses.
Parameter: A nonnegative integer k.
Question: Decide whether $\mathrm{sat}(F) \geq m/2 + k$.

Mahajan and Raman proved that SAT-A$(m/2)$ is fixed-parameter tractable by obtaining a problem kernel with at most $6k + 3$ variables and $10k$ clauses.

We improve on this by obtaining a kernel with at most $4k$ variables and $8.47k$ clauses.

Since $\hat{\varphi}m$ rather than $m/2$ is an asymptotically tight lower bound for UCF CNF formulae, Mahajan and Raman [14] also introduced the following parameterization of Max-SAT:

> SAT-A($\hat{\varphi}m$)
> *Instance:* A UCF CNF formula F with m clauses.
> *Parameter:* A nonnegative integer k.
> *Question:* Decide whether $\mathrm{sat}(F) \geq \hat{\varphi}m + k$.

Mahajan and Raman conjectured that SAT-A($\hat{\varphi}m$) is fixed-parameter tractable. To solve the conjecture in the affirmative, we show the existence of an $O(k)$-variable kernel for SAT-A($\hat{\varphi}m$). This result follows from our improvement of the Lieberherr-Specker lower bound.

The rest of this paper is organized as follows. In Section 2, we give further terminology and notation and some basic results. Section 3 proves the improvement of the Lieberherr-Specker lower bound on $\mathrm{sat}(F)$ assuming correctness of the following lemma: if $F = (V, C)$ is a compact CNF formula, then $\mathrm{sat}(F) \geq \hat{\varphi}|C| + (2 - 3\hat{\varphi})|V|/2$ (we give definition of a compact CNF formula in the next section). We prove this non-trivial lemma in Section 4. In Section 5 we solve the conjecture of Mahajan and Raman [14] in the affirmative and improve their result on SAT-A($m/2$). We conclude the paper with discussions and open problems.

2 Additional Terminology, Notation and Basic Results

We let $F = (V, C)$ denote a CNF formula with a set of variables V and a multiset of clauses C. It is normally assumed that each clause may appear multiple times in C. For the sake of convenience, we assume that each clause appears at most once, but allow each clause to have an integer *weight*. (Thus, instead of saying a clause c appears t times, we will say that c has weight t). If at any point a particular clause c appears more than once in C, we replace all occurrences of c with a single occurrence of the same total weight. We use $w(c)$ to denote the weight of a clause c. For any clause $c \notin C$ we set $w(c) = 0$. If $C' \subseteq C$ is a subset of clauses, then $w(C')$ denotes the sum of the weights of the clauses in C'. For a formula $F = (V, C)$ we will often write $w(F)$ instead of $w(C)$.

For a formula $F = (V, C)$ and a subset $U \subseteq V$ of variables, F_U denotes the subformula of F obtained from F by deleting all clauses without variables in U.

For a CNF formula $F = (V, C)$, a *truth assignment* is a function $\alpha : V \to \{\text{TRUE}, \text{FALSE}\}$. A truth assignment α *satisfies* a clause c if there exists $x \in V$ such that $x \in c$ and $\alpha(x) = \text{TRUE}$, or $\bar{x} \in c$ and $\alpha(x) = \text{FALSE}$. The *weight* of a truth assignment is the sum of the weights of all clauses satisfied by the

assignment. The maximum weight of a truth assignment for F is denoted by $\mathrm{sat}(F)$.

A function $\beta : U \to \{\text{TRUE}, \text{FALSE}\}$, where U is a subset of V is called a *partial truth assignment*. A partial truth assignment $\beta : U \to \{\text{TRUE}, \text{FALSE}\}$ is an *autarky* if β satisfies all clauses of F_U. Autarkies are of interest, in particular, due to the following simple fact whose trivial proof is omitted.

Lemma 1. *Let $\beta : U \to \{\text{TRUE}, \text{FALSE}\}$ be an autarky for a CNF formula F. Then* $\mathrm{sat}(F) = w(F_U) + \mathrm{sat}(F \setminus F_U)$.

A version of Lemma 1 can be traced back to Monien and Speckenmeyer [15].

Recall that a formula $F = (V, C)$ is called *expanding* if $|X| \leq w(F_X)$ for each $X \subseteq V$. We associate a bipartite graph with a CNF formula $F = (V, C)$ as follows: the *bipartite graph B_F of F* has partite sets V and C and the edge vc is in B_F if and only if the variable v or its negation $\bar{v}$ appears in the clause c. Later we will make use of the following result which is a version of Hall's Theorem on matchings in bipartite graphs (cf. [20]).

Lemma 2. *The bipartite graph B_F has a matching covering V if and only if F is expanding.*

We call a CNF formula $F = (V, C)$ *compact* if the following conditions hold:

1. All clauses in F have the form (x) or $(\bar{x} \vee \bar{y})$ for some $x, y \in V$.
2. For every variable $x \in V$, the clause (x) is in C.

A *parameterized problem* is a subset $L \subseteq \Sigma^* \times \mathbb{N}$ over a finite alphabet Σ. L is *fixed-parameter tractable* if the membership of an instance (I, k) in $\Sigma^* \times \mathbb{N}$ can be decided in time $f(k)|I|^{O(1)}$ where f is a function of the *parameter* k only [4, 5, 16]. Given a parameterized problem L, a *kernelization of L* is a polynomial-time algorithm that maps an instance (x, k) to an instance (x', k') (the *kernel*) such that (i) $(x, k) \in L$ if and only if $(x', k') \in L$, (ii) $k' \leq h(k)$, and (iii) $|x'| \leq g(k)$ for some functions h and g. It is well-known [4, 5, 16] that a decidable parameterized problem L is fixed-parameter tractable if and only if it has a kernel. By replacing Condition (ii) in the definition of a kernel by $k' \leq k$, we obtain a definition of a *proper kernel* (sometimes, it is called a *strong kernel*); cf. [1, 3].

3 New Lower Bound for $\mathrm{sat}(F)$

We would like to prove a lower bound on $\mathrm{sat}(F)$ that includes the number of variables as a factor. It is clear that for general CNF formula F such a bound is impossible. For consider a formula containing a single clause c containing a large number of variables. We can arbitrarily increase the number of variables in the formula, and the maximum number of satisfiable clauses will always be 1. We therefore need a reduction rule that cuts out 'excess' variables. Our reduction rule is based on the following lemma proved in Fleischner et al. [6] (Lemma 10) and Szeider [17] (Lemma 9).

Lemma 3. *Let $F = (V, C)$ be a CNF formula. Given a maximum matching in the bipartite graph B_F, in time $O(|C|)$ we can find an autarky $\beta : U \to \{\text{TRUE}, \text{FALSE}\}$ such that $|X| + 1 \leq w(F_X)$ for every $X \subseteq V \backslash U$.*

Note that the autarky found in Lemma 3 can be empty, i.e., $U = \emptyset$. An autarky found by the algorithm of Lemma 3 is of a special kind, called a matching autarky; such autarkies were used first by Aharoni and Linial [2]. Results similar to Lemma 3 have been obtained in the parameterized complexity literature as well, see, e.g., [13].

Lemmas 1 and 3 immediately imply the following:

Theorem 1. *[6, 17] Let F be a CNF formula and let $\beta : U \to \{\text{TRUE}, \text{FALSE}\}$ be an autarky found by the algorithm of Lemma 1. Then $\text{sat}(F) = \text{sat}(F \setminus F_U) + w(F_U)$ and $F \setminus F_U$ is an expanding formula.*

Our improvement of the Lieberherr-Specker lower bound on $\text{sat}(F)$ for a UCF CNF formula F will follow immediately from Theorems 1 and 2 (stated below). It is much harder to prove Theorem 2 than Theorem 1, and our proof of Theorem 2 is based on the following quite non-trivial lemma that will be proved in the next section.

Lemma 4. *If $F = (V, C)$ is a compact CNF formula, then there exists a truth assignment with weight at least*

$$\hat{\varphi} w(C) + \frac{|V|(2 - 3\hat{\varphi})}{2},$$

where $\hat{\varphi} = (\sqrt{5} - 1)/2$, and such an assignment can be found in polynomial time.

The next proof builds on some of the basic ideas in [11].

Theorem 2. *If $F = (V, C)$ is an expanding UCF CNF formula, then there exists a truth assignment with weight at least*

$$\hat{\varphi} w(C) + \frac{|V|(2 - 3\hat{\varphi})}{2},$$

where $\hat{\varphi} = (\sqrt{5} - 1)/2$ and such an assignment can be found in polynomial time.

Proof. We will describe a polynomial-time transformation from F to a compact CNF formula F', such that $|V'| = |V|$ and $w(C') = w(C)$, and any truth assignment for F' can be turned into truth assignment for F of greater or equal weight. The theorem then follows from Lemma 4.

By Lemma 2, there is a matching in the bipartite graph B_F covering V. For each $x \in V$ let c_x be the unique clause associated with x in this matching. For each variable x, if the unit clause (x) or $(\bar{x})$ appears in C, leave c_x as it is for now. Otherwise, remove all variables except x from c_x. We now have that for every x, exactly one of (x), $(\bar{x})$ appears in C.

If $(\bar{x})$ is in C, replace every occurrence of the literal $\bar{x}$ in the clauses of C with x, and replace every occurrence of x with $\bar{x}$. We now have that Condition

2 in the definition of a compact formula is satisfied. For any clause c which contains more than one variable and at least one positive literal, remove all variables except one that occurs as a positive. For any clause which contains only negative literals, remove all but two variables. We now have that Condition 1 is satisfied. This completes the transformation.

In the transformation, no clauses or variables were completely removed, so $|V'| = |V|$ and $w(C') = w(C)$. Observe that the transformation takes polynomial time, and that any truth assignment for the compact formula F' can be turned into a truth assignment for F of greater or equal weight. Indeed, for some truth assignment for F', flip the assignment to x if and only if we replaced occurrences of x with $\bar{x}$ in the transformation. This gives a truth assignment for F such that every clause will be satisfied if its corresponding clause in F' is satisfied. □

Our main result follows immediately from Theorems 1 and 2.

Theorem 3. *Every UCF CNF formula $F = (V, C)$ contains a (possibly empty) subformula $F' = (V', C')$ that can be found in polynomial time and such that*

$$\mathrm{sat}(F) \geq \hat{\varphi} w(C) + (1 - \hat{\varphi}) w(C') + (2 - 3\hat{\varphi})|V \setminus V'|/2.$$

4 Proof of Lemma 4

In this section, we use the fact that $\hat{\varphi} = (\sqrt{5} - 1)/2$ is the positive root of the polynomial $\hat{\varphi}^2 + \hat{\varphi} - 1$. We call a clause $(\bar{x} \vee \bar{y})$ *good* if for every literal $\bar{z}$, the set of clauses containing $\bar{z}$ is not equal to $\{(\bar{x} \vee \bar{z}), (\bar{y} \vee \bar{z})\}$. We define $w_v(x)$ to be the total weight of all clauses containing the literal x, and $w_v(\bar{x})$ the total weight of all clauses containing the literal $\bar{x}$. (Note that $w_v(\bar{x})$ is different from $w(\bar{x})$, which is the weight of the particular clause $(\bar{x})$.) Let $\epsilon(x) = w_v(x) - \hat{\varphi} w_v(\bar{x})$. Let $\gamma = (2 - 3\hat{\varphi})/2 = (1 - \hat{\varphi})^2/2$ and let $\Delta(F) = \mathrm{sat}(F) - \hat{\varphi} w(C)$.

To prove Lemma 4, we will use an algorithm, Algorithm A, described below. We will show that, for any compact CNF formula $F = (V, C)$, Algorithm A finds a truth assignment with weight at least $\hat{\varphi} w(C) + \gamma|V|$. Step 3 of the algorithm removes any clauses which are satisfied or falsified by the given assignment of truth values to the variables. The purpose of Step 4 is to make sure the new formula is compact.

Algorithm A works as follows. Let F be a compact CNF formula. If F contains a variable x such that we can assign x TRUE and increase sufficiently the average number of satisfied clauses, we do just that (see Cases A and B of the algorithm). Otherwise, to achieve similar effect we have to assign truth values to two or three variables (see Cases C and D). Step 3 of the algorithm removes any clauses which are satisfied or falsified by the given assignment of truth values to the variables. The purpose of Step 4 is to make sure the new formula is compact.

Algorithm A

While $|V| > 0$, repeat the following steps:

1. For each $x \in V$, calculate $w_v(x)$ and $w_v(\bar{x})$.

2. Mark some of the variables as TRUE or FALSE, according to the following cases:

   Case A: *There exists $x \in V$ with $w_v(x) \geq w_v(\bar{x})$.* Pick one such x and assign it TRUE.

   Case B: *Case A is false, and there exists $x \in V$ with $(1-\hat{\varphi})\epsilon(x) \geq \gamma$.* Pick one such x and assign it TRUE.

   Case C: *Cases A and B are false, and there exists a good clause.* Pick such a good clause $(\bar{x} \vee \bar{y})$, with (without loss of generality) $\epsilon(x) \geq \epsilon(y)$, and assign y FALSE and x TRUE.

   Case D: *Cases A, B and C are false.* Pick any clause $(\bar{x} \vee \bar{y})$ and pick z such that both clauses $(\bar{x} \vee \bar{z})$ and $(\bar{y} \vee \bar{z})$ exist. Consider the six clauses $(x), (y), (z), (\bar{x} \vee \bar{y}), (\bar{x} \vee \bar{z}), (\bar{y} \vee \bar{z})$ and all 2^3 assignments to the variables x, y, z, and pick an assignment maximizing the total weight of satisfied clauses among the six clauses.

3. Perform the following simplification: For any variable x assigned FALSE, remove any clause containing $\bar{x}$, remove the unit clause (x), and remove x from V. For any variable x assigned TRUE, remove the unit clause (x), remove $\bar{x}$ from any clause containing $\bar{x}$ and remove x from V.

4. For each y remaining, if there is a clause of the form $(\bar{y})$, do the following: If the weight of this clause is greater than $w_v(y)$, then replace all clauses containing the variable y (that is, literals y or $\bar{y}$) with one clause (y) of weight $w_v(\bar{y}) - w_v(y)$. Otherwise remove $(\bar{y})$ from C and change the weight of (y) to $w(y) - w(\bar{y})$.

In order to show that the algorithm finds a truth assignment with weight at least $\hat{\varphi} w(C) + \frac{|V|(2-3\hat{\varphi})}{2}$, we need the following two lemmas.

Lemma 5. *For a formula F, if we assign a variable x TRUE, and run Steps 3 and 4 of the algorithm, the resulting formula $F^* = (V^*, C^*)$ satisfies*

$$\Delta(F) \geq \Delta(F^*) + (1 - \hat{\varphi})\epsilon(x).$$

Furthermore, we have $|V^*| = |V| - 1$, unless there exists $y \in V^*$ such that (y) and $(\bar{x} \vee \bar{y})$ are the only clauses containing y and they have the same weight. In this case, y is removed from V^*.

Proof. Observe that at Step 3, the clause (x) (of weight $w_v(x)$) is removed, clauses of the form $(\bar{x} \vee \bar{y})$ (total weight $w_v(\bar{x})$) become $(\bar{y})$, and the variable x is removed from V.

At Step 4, observe that for each y such that $(\bar{y})$ is now a clause, $w(C)$ is decreased by $2w_y$ and $\mathrm{sat}(F)$ is decreased by w_y, where $w_y = \min\{w(y), w(\bar{y})\}$. Let $q = \sum_y w_y$, and observe that $q \leq w_v(\bar{x})$. A variable y will only be removed at this stage if the clause $(\bar{x} \vee \bar{y})$ was originally in C. We therefore have

1. $\mathrm{sat}(F^*) \leq \mathrm{sat}(F) - w_v(x) - q$
2. $w(C^*) = w(C) - w_v(x) - 2q$

Using the above, we get

$$\begin{aligned}
\Delta(F) &= \mathrm{sat}(F) - \hat{\varphi} \cdot w(C) \\
&\geq (w_v(x) + \mathrm{sat}(F^*) + q) - \hat{\varphi}(w(C^*) + 2q + w_v(x)) \\
&= \Delta(F^*) + (1 - \hat{\varphi})w_v(x) - (2\hat{\varphi} - 1)q \\
&\geq \Delta(F^*) + (1 - \hat{\varphi})(\epsilon(x) + \hat{\varphi} \cdot w_v(\bar{x})) - (2\hat{\varphi} - 1)w_v(\bar{x}) \\
&= \Delta(F^*) + (1 - \hat{\varphi} - \hat{\varphi}^2)w_v(\bar{x}) + (1 - \hat{\varphi})\epsilon(x) \\
&= \Delta(F^*) + (1 - \hat{\varphi})\epsilon(x).
\end{aligned}$$

□

Lemma 6. *For a formula F, if we assign a variable x FALSE, and run Steps 3 and 4 of the algorithm, the resulting formula $F^{**} = (V^{**}, C^{**})$ has $|V^{**}| = |V| - 1$ and satisfies $\Delta(F) \geq \Delta(F^{**}) - \hat{\varphi}\epsilon(x)$.*

Proof. Observe that at Step 3, every clause containing the variable x is removed, and no other clauses will be removed at Steps 3 and 4. Since the clause (y) appears for every other variable y, this implies that $|V^{**}| = |V| - 1$. We also have the following: $\mathrm{sat}(F^{**}) \leq \mathrm{sat}(F) - w_v(\bar{x})$ and $w(C^{**}) = w(C) - w_v(\bar{x}) - w_v(x)$. Thus,

$$\begin{aligned}
\Delta(F) &= \mathrm{sat}(F) - \hat{\varphi}w(C) \\
&\geq (w_v(\bar{x}) + \mathrm{sat}(F^{**})) - \hat{\varphi}(w(C^{**}) + w_v(\bar{x}) + w_v(x)) \\
&= \Delta(F^{**}) + (1 - \hat{\varphi})w_v(\bar{x}) - \hat{\varphi}w_v(x) \\
&= \Delta(F^{**}) + (1 - \hat{\varphi})w_v(\bar{x}) - \hat{\varphi}(\epsilon(x) + \hat{\varphi} \cdot w_v(\bar{x})) \\
&= \Delta(F^{**}) + (1 - \hat{\varphi} - \hat{\varphi}^2)w_v(\bar{x}) - \hat{\varphi}\epsilon(x) \\
&= \Delta(F^{**}) - \hat{\varphi}\epsilon(x).
\end{aligned}$$

□

Now we are ready to prove Lemma 4.

Proof of Lemma 4: We will show that Algorithm A finds a truth assignment with weight at least $\hat{\varphi}w(C) + \frac{|V|(2-3\hat{\varphi})}{2}$. Note that the inequality in the lemma can be reformulated as $\Delta(F) \geq \gamma|V|$.

Let F and $\hat{\varphi}$ be defined as in the statement of the lemma. Note that at each iteration of the algorithm, at least one variable is removed. Therefore, we will show the lemma by induction on $|V|$. If $|V| = 0$ then we are done trivially and if $|V| = 1$ then we are done as $\mathrm{sat}(F) = w(C) \geq \hat{\varphi}w(C) + \gamma$ (as $w(C) \geq 1$). So assume that $|V| \geq 2$.

For the induction step, let $F' = (V', C')$ be the formula resulting from F after running Steps 1-4 of the algorithm, and assume that $\Delta(F') \geq \gamma|V'|$. We show that $\Delta(F) \geq \gamma|V|$, by analyzing each possible case in Step 2 separately.

Case A: *$w_v(x) \geq w_v(\bar{x})$ for some $x \in V$.* In this case we let x be TRUE, which by Lemma 5 implies the following:

$$\begin{aligned}
\Delta(F) &\geq \Delta(F') + (1-\hat{\varphi})\epsilon(x) \\
&= \Delta(F') + (1-\hat{\varphi})(w_v(x) - \hat{\varphi}w_v(\bar{x})) \\
&\geq \Delta(F') + (1-\hat{\varphi})(w_v(x) - \hat{\varphi}w_v(x)) \\
&= \Delta(F') + (1-\hat{\varphi})^2 w_v(x) \\
&= \Delta(F') + 2\gamma w_v(x).
\end{aligned}$$

If $y \in V \setminus V'$, then either $y = x$ or $(\bar{x} \vee \bar{y}) \in C$. Therefore $|V| - |V'| \leq w_v(\bar{x}) + 1 \leq w_v(x) + 1$. As $w_v(x) \geq 1$ we note that $2\gamma w_v(x) \geq \gamma(w_v(x) + 1)$. This implies the following, by induction, which completes the proof of Case A.

$$\begin{aligned}
\Delta(F) &\geq \Delta(F') + \gamma(w_v(x) + 1) \\
&\geq \gamma|V'| + \gamma(w_v(x) + 1) \geq \gamma|V|.
\end{aligned}$$

Case B: *Case A is false, and $(1-\hat{\varphi})\epsilon(x) \geq \gamma$ for some $x \in V$.*

Again we let x be TRUE. Since $w_v(y) < w_v(\bar{y})$ for all $y \in V$, we have $|V| = |V'| + 1$. Analogously to Case A, using Lemma 5, we get the following:

$$\begin{aligned}
\Delta(F) &\geq \Delta(F') + (1-\hat{\varphi})\epsilon(x) \\
&\geq \gamma|V'| + \gamma = \gamma|V|.
\end{aligned}$$

For Cases C and D, we generate a graph G from the set of clauses. The vertex set of G is the variables in V (i.e. $V(G) = V$) and there is an edge between x and y if and only if the clause $(\bar{x} \vee \bar{y})$ exists in C. A *good* edge in G is an edge $uv \in E(G)$ such that no vertex $z \in V$ has $N(z) = \{u, v\}$ (that is, an edge is good if and only if the corresponding clause is good).

Case C: *Cases A and B are false, and there exists a good clause $(\bar{x} \vee \bar{y})$.* Without loss of generality assume that $\epsilon(x) \geq \epsilon(y)$. We will first let y be FALSE and then we will let x be TRUE. By letting y be FALSE we get the following by Lemma 6, where F^{**} is defined in Lemma 6: $\Delta(F) \geq \Delta(F^{**}) - \hat{\varphi}\epsilon(x)$.

Note that the clause $(\bar{x} \vee \bar{y})$ has been removed so $w_v^{**}(\bar{x}) = w_v(\bar{x}) - w(\bar{x} \vee \bar{y})$ and $w_v^{**}(x) = w_v(x)$ (where $w_v^{**}(.)$ denote the weights in F^{**}). Therefore using Lemma 5 on F^{**} instead of F we get the following, where the formula F^* in Lemma 5 is denoted by F' below and $w^0 = w(\bar{x} \vee \bar{y})$:

$$\Delta(F^{**}) \quad \geq \quad \Delta(F') + (1-\hat{\varphi})(w_v(x) - \hat{\varphi}(w_v(\bar{x}) - w^0)).$$

First we show that $|V'| = |V^{**}| - 1 = |V| - 2$. Assume that $z \in V \setminus (V' \cup \{x, y\})$ and note that $N(z) \subseteq \{x, y\}$. Clearly $|N(z)| = 1$ as xy is a good edge. If $N(z) = \{y\}$ then $(z) \in C'$, so we must have $N(z) = \{x\}$. However the only way $z \notin V'$ is if $w_v(z) = w_v(\bar{z})$, a contradiction as Case A is false. Therefore, $|V'| = |V| - 2$, and the following holds by the induction hypothesis.

$$
\begin{array}{rcl}
\Delta(F) & \geq & \Delta(F^{**}) - \hat{\varphi}\epsilon(x) \\
& \geq & \Delta(F') + (1 - \hat{\varphi})(w_v(x) - \hat{\varphi}(w_v(\bar{x}) - w^0)) - \hat{\varphi}\epsilon(x) \\
& \geq & \gamma|V'| + (1 - \hat{\varphi})(\epsilon(x) + \hat{\varphi}w^0) - \hat{\varphi}\epsilon(x) \\
& = & \gamma|V| - 2\gamma + (1 - \hat{\varphi})\hat{\varphi}w^0 - (2\hat{\varphi} - 1)\epsilon(x).
\end{array}
$$

We would be done if we can show that $2\gamma \leq (1 - \hat{\varphi})\hat{\varphi}w^0 - (2\hat{\varphi} - 1)\epsilon(x)$. As $w^0 \geq 1$ and we know that, since Case B does not hold, $(1 - \hat{\varphi})\epsilon(x) < \gamma$, we would be done if we can show that $2\gamma \leq (1 - \hat{\varphi})\hat{\varphi} - (2\hat{\varphi} - 1)\gamma/(1 - \hat{\varphi})$. This is equivalent to $\gamma = (1 - \hat{\varphi})^2/2 \leq \hat{\varphi}(1 - \hat{\varphi})^2$, which is true, completing the proof of Case C.

Case D: *Cases A, B and C are false.* Then G has no good edge.

Assume xy is some edge in G and $z \in V$ such that $N(z) = \{x, y\}$. As xz is not a good edge there exists a $v \in V$, such that $N(v) = \{x, z\}$. However v is adjacent to z and, thus, $v \in N(z) = \{x, y\}$, which implies that $v = y$. This shows that $N(y) = \{x, z\}$. Analogously we can show that $N(x) = \{y, z\}$. Therefore, the only clauses in C that contain a variable from $\{x, y, z\}$ form the following set: $S = \{(x), (y), (z), (\bar{x} \vee \bar{y}), (\bar{x} \vee \bar{z}), (\bar{y} \vee \bar{z})\}$.

Let F' be the formula obtained by deleting the variables x, y and z and all clauses containing them. Now consider the three assignments of truth values to x, y, z such that only one of the three variables is assigned False. Observe that the total weight of clauses satisfied by these three assignments equals

$$w_v(\bar{x}) + w_v(\bar{y}) + w_v(\bar{z}) + 2(w(x) + w(y) + w(z)) = 2W,$$

where W is the total weight of the clauses in S. Thus, one of the three assignments satisfies the weight of at least $2W/3$ among the clauses in S. Observe also that $w(C) - w(C') \geq 6$, and, thus, the following holds.

$$
\begin{array}{rcl}
\Delta(F) & \geq & 2(w(C) - w(C'))/3 + \mathrm{sat}(F') - \hat{\varphi}(w(C') + w(C) - w(C')) \\
& \geq & \gamma|V'| + 2(w(C) - w(C'))/3 - \hat{\varphi}(w(C) - w(C')) \\
& = & \gamma|V| - 3\gamma + (2 - 3\hat{\varphi})(w(C) - w(C'))/3 \\
& \geq & \gamma|V| - 3\gamma + 2(2 - 3\hat{\varphi}) > \gamma|V|.
\end{array}
$$

This completes the proof of the correctness of Algorithm A. It remains to show that Algorithm A takes polynomial time.

Each iteration of the algorithm takes $O(nm)$ time. The algorithm stops when V is empty, and at each iteration some variables are removed from V. Therefore, the algorithm goes through at most n iterations and, in total, it takes $O(n^2m)$ time. This completes the proof of Lemma 4. □

Note that the bound $(2-3\hat{\varphi})/2$ in Lemma 4 cannot be improved due to the following example. Let l be any positive integer and let $F = (V, C)$ be defined such that $V = \{x_1, x_2, \dots, x_l, y_1, y_2, \dots, y_l\}$ and C contain the constraints (x_1), (x_2),..., (x_l), (y_1), (y_2),..., (y_l) and $(\bar{x}_1 \vee \bar{y}_1)$, $(\bar{x}_2 \vee \bar{y}_2)$,..., $(\bar{x}_l \vee \bar{y}_l)$. Let the weight of every constraint be one and note that for every i we can only satisfy two of the three constraints (x_i), (y_i) and $(\bar{x}_i \vee \bar{y}_i)$. Therefore $\text{sat}(F) = 2l$ and the following holds:

$$\hat{\varphi}w(C) + \tfrac{|V|(2-3\hat{\varphi})}{2} = 3l\hat{\varphi} + \tfrac{2l(2-3\hat{\varphi})}{2} = l(3\hat{\varphi} + 2 - 3\hat{\varphi}) = 2l = \text{sat}(F).$$

5 Parameterized Complexity Results

Recall that formulations of parameterized problems SAT-A$(m/2)$ and SAT-A$(\hat{\varphi}m)$ were given in Section 1.

Theorem 4. *The problem* SAT-A$(\hat{\varphi}m)$ *has a proper kernel with at most* $\lfloor(7+3\sqrt{5})k\rfloor$ *variables.*

Proof. Consider an instance $(F = (V, C), k)$ of the problem. By Theorem 1, there is an autarky $\beta : U \to \{\text{TRUE}, \text{FALSE}\}$ which can be found by the polynomial algorithm of Lemma 1 such that $\text{sat}(F) = \text{sat}(F \setminus F_U) + w(F_U)$ and $F \setminus F_U$ is an expanding formula.

If $U = V$, then $\text{sat}(F) = w(F)$, and the kernel is trivial.

Now suppose that $U \neq V$ and denote $F \setminus F_U$ by $F' = (V', C')$. We want to choose an integral parameter k' such that (F, k) is a YES-instance of the problem if and only if (F', k') is a YES-instance of the problem. It is enough for k' to satisfy $\text{sat}(F) - \lfloor\hat{\varphi}w(F)\rfloor - k = \text{sat}(F') - \lfloor\hat{\varphi}w(F')\rfloor - k'$. By Theorem 1, $\text{sat}(F') = \text{sat}(F) - w(F) + w(F')$. Therefore, we can set $k' = k - w(F) + w(F') + \lfloor\hat{\varphi}w(F)\rfloor - \lfloor\hat{\varphi}w(F')\rfloor$. Since $w(F) - w(F') \geq \lceil\hat{\varphi}(w(F) - w(F'))\rceil \geq \lfloor\hat{\varphi}w(F)\rfloor - \lfloor\hat{\varphi}w(F')\rfloor$, we have $k' \leq k$.

By Theorem 2, if $k' \leq \frac{|V'|(2-3\hat{\varphi})}{2}$, then F is a YES-instance of the problem. Otherwise, $|V'| < \frac{2k}{2-3\hat{\varphi}} = (7+3\sqrt{5})k$. Note that F' is not necessarily a kernel as $w(F')$ is not necessarily bounded by a function of k. However, if $w(F') \geq 2^{2k/(2-3\hat{\varphi})}$ then we can solve the instance (F', k') in time $O(w(F')^2)$ and, thus, we may assume that $w(F') < 2^{2k/(2-3\hat{\varphi})}$, in which case, F' is the required kernel. □

Theorem 5. *The problem* SAT-A$(m/2)$ *has a proper kernel with at most* $4k$ *variables and* $(2\sqrt{5}+4)k \leq 8.473k$ *clauses.*

Proof. First, we reduce the instance to a UCF instance $F = (V, C)$. As in Theorem 4, in polynomial time, we can obtain an expanding formula $F' = (V', C')$. Again, we want to choose a parameter k' such that (F, k) is a YES-instance if and only if (F', k') is a YES-instance.

It is enough for k' to satisfy $\mathrm{sat}(F)-\lfloor w(F)/2\rfloor-k=\mathrm{sat}(F')-\lfloor w(F')/2\rfloor-k'$. By Theorem 1, $\mathrm{sat}(F')=\mathrm{sat}(F)-w(F)+w(F')$. Therefore, we can set $k'=k-\lceil w(F)/2\rceil+\lceil w(F')/2\rceil$. As $w(F')\le w(F)$, we have $k'\le k$.

By Theorem 2, there is a truth assignment for F' with weight at least $\hat{\varphi}w(F')+\frac{|V'|(2-3\hat{\varphi})}{2}$. Hence, if $k'\le(\hat{\varphi}-1/2)w(F')+\frac{|V'|(2-3\hat{\varphi})}{2}$, the instance is a YES-instance. Otherwise,

$$k'-\frac{|V'|(2-3\hat{\varphi})}{2}>(\hat{\varphi}-\frac{1}{2})w(F'). \tag{1}$$

The weaker bound $k'>(\hat{\varphi}-\frac{1}{2})w(F')$ is enough to give us the claimed bound on the total weight (i.e., the number) of clauses. To bound the number of variables, note that since F' is expanding, we can satisfy at least $|V'|$ clauses. Thus, if $w(F')/2+k'\le|V'|$, the instance is a YES-instance. Otherwise, $w(F')/2+k'>|V'|$ and

$$2(\hat{\varphi}-\frac{1}{2})(|V'|-k')<(\hat{\varphi}-\frac{1}{2})w(F'). \tag{2}$$

Combining Inequalities (1) and (2), we obtain:

$$2(\hat{\varphi}-\frac{1}{2})(|V'|-k')<(\hat{\varphi}-\frac{1}{2})w(F')<k'-\frac{|V'|(2-3\hat{\varphi})}{2}.$$

This simplifies to $|V'|<4k'\le 4k$, giving the required kernel. □

6 Discussion

A CNF formula I is *t-satisfiable* if any subset of t clauses of I can be satisfied simultaneously. In particular, a CNF formula is unit-conflict free if and only if it is 2-satisfiable. Let r_t be the largest real such that in any t-satisfiable CNF formula at least r_t-th fraction of its clauses can be satisfied simultaneously. Note that $r_1=1/2$ and $r_2=(\sqrt{5}-1)/2$. Lieberherr and Specker [12] and, later, Yannakakis [19] proved that $r_3\ge 2/3$. Käppeli and Scheder [9] proved that $r_3\le 2/3$ and, thus, $r_3=2/3$. Král [10] established the value of r_4: $r_4=3/(5+[(3\sqrt{69}-11)/2]^{1/3}-[3\sqrt{69}+11)/2]^{1/3})\approx 0.6992$.

For general t, Huang and Lieberherr [8] showed that $\lim_{t\to\infty}r_t\le 3/4$ and Trevisan [18] proved that $\lim_{t\to\infty}r_t=3/4$ (a different proof of this result is later given by Král [10]).

In the preliminary version of this paper published in the proceedings of IPEC 2010 we asked to establish parameterized complexity of the following parameterized problem: given a 3-satisfiable CNF formula $F=(V,C)$, decide whether $\mathrm{sat}(F)\ge 2|C|/3+k$, where k is the parameter. This question was recently solved in [7] by showing that the problem has a kernel with a linear number of variables. Unlike this paper, [7] uses the Probabilistic Method. Similar question for any fixed $t>3$ remains open.

References

[1] F.N. Abu-Khzam and H. Fernau, Kernels: Annotated, proper and induced. *Proc. IWPEC 2006*, Lect. Notes Comput. Sci. 4169:264–275, 2006.

[2] R. Aharoni and N. Linial, Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *J. Comb. Theory, Ser. A*, 43(2):196–204, 1986.

[3] Y. Chen, J. Flum, and M. Müller, Lower bounds for kernelizations and other preprocessing procedures. *Proc. CiE 2009*, Lect. Notes Comput. Sci. 5635: 118–128, 2009.

[4] R. G. Downey and M. R. Fellows. *Parameterized Complexity*, Springer, 1999.

[5] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer, 2006.

[6] H. Fleischner, O. Kullmann and S. Szeider, Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference. *Theoret. Comput. Sci.*, 289(1):503–516, 2002.

[7] G. Gutin, M. Jones and A. Yeo, A New Bound for 3-Satisfiable MaxSat and its Algorithmic Application. To appear in *Proc. FCT 2011*.

[8] M.A. Huang and K.J. Lieberherr, Implications of forbidden structures for extremal algorithmic problems. *Theoret. Comput. Sci.* 40: 195–210, 1985.

[9] C. Käppeli and D. Scheder, Partial satisfaction of k-satisfiable formulas. *Electronic Notes in Discrete Math.* 29:497–501, (2007).

[10] D. Král, Locally satisfiable formulas. *Proc. SODA 2004*, 330–339, 2004.

[11] K.J. Lieberherr and E. Specker, Complexity of partial satisfaction. *J. ACM* 28(2):411-421, 1981.

[12] K.J. Lieberherr and E. Specker, Complexity of partial satisfaction, II. Tech. Report 293 of Dept. of EECS, Princeton Univ., 1982.

[13] D. Lokshtanov, *New Methods in Parameterized Algorithms and Complexity*, PhD thesis, Bergen, April 2009.

[14] M. Mahajan and V. Raman. *Parameterizing above guaranteed values: MaxSat and MaxCut. J. Algorithms* 31(2):335–354, 1999.

[15] B. Monien and E. Speckenmeyer. Solving satisfiability in less than 2^n steps. *Discr. Appl. Math.* 10:287–295, 1985.

[16] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.

[17] S. Szeider, Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable. *J. Comput. Syst. Sci.*, 69(4):656–674, 2004.

[18] L. Trevisan. On local versus global satisfiability. *SIAM J. Discret. Math.* 17(4):541–547, 2004.

[19] M. Yannakakis. On the approximation of maximum satisfiability. *J. Algorithms*, 17:475–502, 1994.

[20] D.B. West, *Introduction to Graph Theory*, 2nd Ed., Prentice Hall, 2001.